REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30 Nov 2001	Final 24 Nov 1999 – 30 Sep 2000

4. TITLE AND SUBTITLE

EPBST Workstation Improvement: Automated Point Matching

5. FUNDING NUMBERS

6. AUTHORS

Stephen J. Dow

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The University of Alabama in Huntsville
Huntsville, AL 35899

8. PERFORMING ORGANIZATION REPORT NUMBER

5-20801

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Aviation & Missile Command
Redstone Arsenal, AL 35898

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

As part of the workflow of the EPBST workstation software for building terrain models, point correspondences (x1, y1), (x2, y2), representing the locations of a common physical feature in two images, must be identified. Early versions of the software required that all such point correspondences be identified manually. This report describes the methodology used in adding automated point matching to the software, using (1) interest point detection, (2) correlation, (3) consistency checks, and (4) subpixel refinement by least squares adjustment.

14. SUBJECT TERMS

Image matching, correlation, interest point, least squares

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

PLEASE CHECK THE APPROPRIATE BLOCK BELOW

DAO# _____

☐ _____ copies are being forwarded. Indicate whether Statement A, B, C, D, E, F, or X applies.

☑ DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

☐ DISTRIBUTION STATEMENT B:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES ONLY; (indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT C:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTS (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT D:
DISTRIBUTION AUTHORIZED TO DoD AND U.S. DoD CONTRACTORS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT E:
DISTRIBUTION AUTHORIZED TO DoD COMPONENTS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT F:
FUTHER DISSEMINATION ONLY AS DIRECTED BY (Indicate Controlling DoD Office and Date) or HIGHER DoD AUTHORITY.

☐ DISTRIBUTION STATEMENT X:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE INDIVIDUALS OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT-CONTROLLED TECHNICAL DATA IN ACCORDANCE WITH DoD DIRECTIVE 5230.25. WITHHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC DISCLOSURE, 6 Nov 1984 (indicate date of determination). CONTROLLING DoD OFFICE IS (Indicate Controlling DoD Office).

☐ This document was previously forwarded to DTIC on _____ (date) and the AD number is _____.

☐ In accordance with provisions of DoD instructions. The document requested is not supplied because:

☐ It will be published at a later date. (Enter approximate date, if known).

☐ Other. (Give Reason)

DoD Directive 5230.24, "Distribution Statements on Technical Documents," 18 Mar 87, contains seven distribution statements, as described briefly above. Technical Documents must be assigned distribution statements.

Stephen J. Dow
Print or Type Name

Stephen J Dow
Authorized Signature/Date

256-824-6406
Telephone Number

EPBST Workstation Improvement: Automated Point Matching

Stephen J. Dow
Department of Mathematical Sciences
The University of Alabama in Huntsville

November 11, 2001

Final Report

F/DOD/ARMY/AMCOM/EPBST Workstation Improvement Effort
DAAH01-97-D-R005 D.O. 22

Period of Performance: 11/24/99 to 9/30/00

1. Introduction

The EPBST workstation is a set of software tools used to create terrain models and target paths for use in the Javelin Enhanced Producibility Basic Skills Trainer (EPBST) and other weapons trainers. The construction of each terrain model requires stitching photographs to create a panoramic image in the visible spectrum, and creating a corresponding image in the infrared sprectrum registered to the visible image. A basic step in image registration is solving the image matching problem: given two images having overlapping fields-of-view, identify pairs of points (x1, y1), (x2, y2) representing the locations of a common physical feature in the two images. In the early versions of the EPBST workstation software, such point correspondences could only be identified manually; to identify a point correspondence the user would view the two images in side-by-side windows, and using the mouse click on position (x1, y1) in the first window and then position (x2, y2) in the second window. More recent versions of the software have incorporated automated point matching, to both reduce the work involved and increase the accuracy of these correspondences. This report focuses on the methodology and software developed during the performance of this task for automated point matching.

Since the early 1970's numerous research papers have focused on the image matching problem, and our work is mostly an implementation of previously published methods. Our approach involves the following component processes: (1) interest point detection, (2) correlation, (3) consistency checks, and (4) subpixel refinement by least squares adjustment. The overall approach for the first three of these processes bears some resemblance to that given by Zhang et. al. [4]. The use of correlation relies on the assumption that the two photos were taken with similar camera parameters, so that they are at nearly the same scale, and that any rotation about the line-of-sight is small.

2. Correlation

In this section we introduce correlation in a general mathematical setting, as a measure of vector similarity. Let $\mathbf{u} = (a_1, a_2, ..., a_n)$ and $\mathbf{v} = (b_1, b_2, ..., b_n)$ be two vectors of real numbers, each with **n** components. Let $Sa = \Sigma\, a_i$, the sum of the n components of **u**. Similarly let Sb, Sab, Saa, and Sbb denote, respectively, the sums of the expressions b_i , $a_i b_i$, a_i^2 and b_i^2. The correlation of **u** and **v** may then be computed as

$$\rho(\mathbf{u}, \mathbf{v}) = (n\,Sab - Sa\,Sb) / ((n\,Saa - (Sa)^2)(n\,Sbb - (Sb)^2)^{1/2}.$$

We assign $\rho(\mathbf{u}, \mathbf{v})$ the value 0 when the expression in the denominator is 0.

The correlation function has the following properties:

- $-1 \le \rho(\mathbf{u}, \mathbf{v}) \le 1$ for all **u**, **v**
- $\rho(\mathbf{u}, \mathbf{v}) = \rho(\mathbf{v}, \mathbf{u})$
- $\rho(\mathbf{u}, \mathbf{v}) = \rho(c\mathbf{u} + \mathbf{d}, \mathbf{v})$ where c is any positive number and $\mathbf{d} = (d, d, ..., d)$, a constant vector.

Intuitively, $\rho(\mathbf{u}, \mathbf{v})$ is close to 1 when **u**, **v** have similar patterns of increase and decrease; at any point along their length, they are both increasing or both decreasing. For example,

$$\rho((0, 1, 1, 2, 2), (2, 3, 4, 4, 4)) = 0.869,$$

with both vectors are generally increasing throughout, and

$$\rho((0, 1, 3, 2, 0), (5, 8, 9, 6, 3)) = 0.787,$$

with both vectors increasing up to their third component and then decreasing. When two vectors have opposite patterns, one increasing whenever the other is decreasing, their correlation will be close to –1; when there is no common pattern their correlation will be close to 0.

The examples above illustrate that high correlation depends only on similarity of the numeric patterns and not on the absolute differences between corresponding components of the vectors; this is also evident from the third property above. To measure absolute differences, we might use the measure

$$(\Sigma\,(a_i - b_i)^2/n)^{1/2}$$

which is simply the square root of the average squared difference of corresponding components. We have found that this measure goes to the other extreme in that any difference in the means of **u** and **v** dominates the measure, overwhelming its ability to pick out whatever similarity in pattern exists. As an attempt at an intermediate between measuring patterns and absolute differences we have obtained some good results with

$$\delta(\mathbf{u}, \mathbf{v}) = (\Sigma\,(a_i - b_i)^2/n - 0.5(Sa/n - Sb/n)^2)^{1/2}$$

which subtracts out half the squared difference of means. If the vector components all lie between 0 and some maximum value M (for example M = 255 for components representing 8-bit pixel values), then δ satisfies $0 \le \delta(\mathbf{u}, \mathbf{v}) \le M$ and we can divide δ by M to get a measure with maximum value 1, as we have with ρ.

So far we have described correlation as applied to one-dimensional vectors of real numbers. Image correlation is performed by applying the correlation function to two-dimensional pixel neighborhoods from each of two images. Let f represent an image, with f(x, y) the pixel value at a given

pixel location (x, y). To measure the similarity of image f in the vicinity of pixel (x1, y1) and image g in the vicinity of (x2, y2), we compute $\rho(\mathbf{u}, \mathbf{v})$, for vectors **u** and **v** formed from corresponding pixel values surrounding (x1, y1) and (x2, y2) respectively. For example, if we use 3 x 3 pixel neighborhoods vectors **u** and **v** will each have 9 components, as follows:

$$\mathbf{u} = (f(x1 - 1, y1 - 1), f(x1, y1 - 1), \ldots, f(x1 + 1, y1 + 1))$$

$$\mathbf{v} = (g(x1 - 1, y1 - 1), g(x1, y1 - 1), \ldots, g(x1 + 1, y1 + 1))$$

In general, the neighborhoods have size (2k + 1) x (2k + 1), and we denote the correlation between f and g at a given point pair as $\rho(f, x1, y1, g, x2, y2, k)$.

3. Interest Point Detection

Interest point detection seeks out candidates points from each of the two images from which the subsequent steps will search for corresponding pairs. This initial step reduces the computational load, as well as the chance of errant matches, by limiting the remaining processes to a discrete set of points, chosen to have good matching characteristics. The principle desirable characteristic of an interest point is its distinctiveness from neighboring points, which has the effect of making it less likely that the point will closely match multiple locations in the other image. For example, consider a straight edge between light and dark regions which appears in both images, such as the edge of a straight road. In general, a point along that edge is not a good interest point, because it will be well-matched to various points along the corresponding edge in the other image. On the other hand, a point along the edge where a unique notch or mark occurs would likely be a good interest point.

Interest point detection begins with an operator which assigns a numerical value to each pixel in an image, and then proceeds to select pixels from this output as interest points according to some criteria, typically involving thresholding and nonmaxima suppression. In our implementation, the latter step involves a "distributed points search" described below. We experimented with three types of interest point operators, and settled on the third for incorporation into the workstation software:

(1) Autocorrelation: Operators of this type search for points where the image correlates poorly with any shifted version of itself. One form of this operator, applied to an image f, is

$$I(x, y) = 1 - \max_{i = 1..8} \rho(f, x, y, f, x_i, y_i, 1)$$

where (x_i, y_i), i = 1, ..., 8 are the eight pixel locations surrounding pixel (x, y).

(2) Harris Operator: This operator takes the form

$$I(x, y) = \det(A) - 0.04(\text{trace}(A))^2, \text{ where}$$

$$A = \begin{pmatrix} S(g_x^2) & S(g_x g_y) \\ S(g_x g_y) & S(g_y^2) \end{pmatrix}$$

and g_x and g_y are differences of the input image in the x and y directions, and S is a smoothing operator. This operator has been widely used since its introduction in the late 1980's, for example, it was chosen by Zhang et. al. [4] for the interest point detection stage of their matching algorithm. An operator based on the same principles was independently developed by Forstner [1].

(3) Minimum Intensity Change: This operator, introduced by Trajkovic and Hedley [3], takes the form

$$I(x,y) = \max_{i=1..k} ((f(x + x_i, y + y_i) - f(x, y))^2 + (f(x - x_i, y - y_i) - f(x, y))^2),$$

where the 2k points $(x + x_i, y + y_i)$, $(x - x_i, y - y_i)$, i = 1..k, form a "digital circle" about (x, y). In our implementation we used k = 6, and (x_i, y_i) = (2, -1), (2, 0), (2, 1), (1, 2), (0, 2), (-1, 2), as shown below

	q_4	q_5	q_6	
q_3				p_1
q_2		p		p_2
q_1				p_3
	p_6	p_5	p_4	

where p = (x, y), $p_i = (x + x_i, y + y_i)$, and $q_i = (x - x_i, y - y_i)$.

Each of the preceding operators creates a new image from which we select high-valued pixels as interest points. For our purposes, we would like a set of interest points that are well-distributed throughout the image, or throughout a rectangular subimage of our choosing. (This subregion could be the region we expect through initial estimates to be the overlap area with the other image.) To perform the selection, we first compute a threshold value T such that I(x, y) > T for a specified percentage (e.g. 1%) of the pixels. We also specify a fixed number of desired interest points in advance, say M, where M is normally considerably smaller than the percentage of pixels satisfying the threshold, so that the "well-distributed" criteria may be imposed. To select the final points, we then tile the input image (or subimage) into an n × n grid of subrectangles ("cells"), where n is the smallest integer such that $n^2 > 4M$. The cells are as uniformly-sized as possible; i.e. if the dimensions of the input image are x × y, then each cell has dimensions approximately x/n × y/n (but they have boundaries on the integer pixel grid and x or y may not be evenly divisible by n). Within each cell we find the pixel having the maximum interest operator output, and if that output exceeds the threshold T, and is a local maximum (no neighboring pixel, in the current cell or another, has a larger value), then it is included in a list of interest points. The list so constructed is sorted by interest operator value, and the top M pixels selected. The choice of the threshold T and of the factor 4 in the condition $n^2 > 4M$ are heuristic. The threshold controls the minimum quality of the interest points selected. The factor affects how well-distributed the points are.

4. Generating Matches by Correlation

We now discuss the generation of matches; i.e. point pairs (x1, y1), (x2, y2), where (x1, y1) is a pixel location in the first image f, and (x2, y2) the corresponding location in the second image g. A central function in the process, called CorrSearch in the code, takes as input a point (x1, y1), and searches through a specified rectangle in the second image for a point (x2, y2) which maximizes ρ(f, x1, y1, g, x2, y2, k). The search is restricted to those points (x2, y2) with interest operator value satisfying a given threshold T. This restriction significantly speeds up the computation, by limiting the number of pixels (x2, y2) for which ρ must be computed. Recall that k determines the size of the pixel neighborhoods used in the computation of ρ. In our implementation we use three successively larger values of k (k = 3, 6, 9) in evaluating ρ,

dropping (x2, y2) as a candidate match for (x1, y1) if any of these values of ρ are below a threshold. This scheme also aids performance. Also we have found that replacing ρ by δ (see section 2) in at least one of these evaluations can help eliminate spurious matches (but have not done a systematic study to test this).

Using the processes described so far, we may build a list of candidate matches as follows. Create a list of interest points in the first image f. Apply the interest operator to the second image g, and compute an interest operator threshold T for g, but do not select interest points. Loop through the interest points of f; for each such point (x1, y1), apply CorrSearch to the whole image g to find the best match (x2, y2) if one exists meeting the criteria described above. If so, store the match (x1, y1, x2, y2) together with its final correlation (or similarity) value (ρ or δ). Finally select the M matches with the highest similarity values, where M is the desired number of matches.

The process just described includes searching the whole image g for each (x2, y2), which is computationally expensive. If additional information is available, such as an approximate 2d transformation from pixel coordinates of f to pixel coordinates of g, then the search for (x2, y2) can be limited to a region about the point where we expect to find it. This additional information can come from previously determined matches, which might have been supplied manually, or by a previous iteration of the search in which a small number of initial matches were generated. For large images the initial search may be performed using reduced resolution versions of the images.

5. Consistency Checks

We now describe methods for choosing, from the list of candidate matches created as in the previous section, a small number of best matches, based on their consistency with one another and support from other matches in the list. Our measures of consistency and support are based on the geometrical relationships among the points.

In the case where the two photos were taken from the same location (the rotational case) with the same camera parameters, the transformation from pixel coordinates to pixel coordinates is a 2d projective transformation:

$$(x', y') = ((a_{11}x + a_{12}y + a_{13})/w,\ (a_{21}x + a_{22}y + a_{23})/w),$$
$$w = a_{31}x + a_{32}y + 1.$$

In the case of two viewing locations (the stereo case), there is no single transformation as in the rotational case, only an epipolar constraint. However, any coplanar set of points are transformed from one image to the other under a common 2d projective transformation. (Here of course the coplanarity refers to the object points corresponding to the matched image points.) In finding initial matches in the stereo case, we assume that at least some four matches are coplanar, with a significant portion of the other matches approximately in the same plane as well. Given four matches, the coefficients a_{ij} may be determined by solving a linear system of 8 equations in 8 unknowns. When just three matches are given, the projective coefficients a_{31}, a_{32} may be taken to be zero and an affine transformation determined:

$$(x', y') = (a_{11}x + a_{12}y + a_{13},\ a_{21}x + a_{22}y + a_{23})$$

With just two matches, a transformation consisting of a rotation, scale, and translation may be determined:

$$(x', y') = r(cx - sy,\ sx + cy) + (t_x, t_y),$$

$$\text{where } c = \cos(\theta), s = \sin(\theta).$$

Our consistency check for a given pair or triplet of matches is that the transformation determined from them be close to a simple translation. In the case of a pair, this is determined by checking that the rotation is small and the scale close to one. In the case of a triplet, for which some pair has already been found consistent, it is checked by verifying that the 2×2 determinant $d = a_{11}\, a_{22} - a_{12}\, a_{21}$ (which may be regarded as the "scale" of the affine transformation) is close to one.

Given a transformation determined from a subset of the matches as above, we measure the support of another match (x1, y1, x2, y2) by how close (x2, y2) is to the image of (x1, y1) under the transformation. The smaller the distance

$$\text{dist}((x2, y2), (x1', y1')) = ((x2 - x1')^2 + (y2 - y1')^2)^{1/2}$$

the better the support. For an overall measure of support for the given subset, we add these distances. However, because we do not want the measure to be corrupted by the nonsupport of false matches, we include only the smallest n_s such distances in the sum, where n_s is the half the number of remaining matches.

These checks and support measures may be used to find the two "best" matches among an initial list of candidate matches as follows. We first remove from the list any match which is inconsistent with all other matches. From the remaining list, we examine all consistent pairs and choose the one with the highest measure of support from the others. To obtain a set of four "best" matches from an initial candidate list, we first select the best two in the manner just described. We then loop through the remaining matches, examining each possible pair of third and fourth matches which are each consistent with the initial two as a triplet. Among such foursomes we take the one with the highest measure of support from the others, as before.

6. Subpixel Refinement by Least Squares Adjustment

The correlation-based methods in section 4 provide integer pixel coordinates for the matches. To further refine the match, we use least squares adjustment to estimate the local transformation from one image to the other. Methods similar to our formulation were reported in various papers during the 1980's, such as Grün [2]. We seek parameters $a_0, a_1, \ldots, a_7$ which minimize the function

$$F = a_0 + a_1 g(u,v) - f(x,y), \text{ where}$$

$$u = a_2 + a_3 x + a_4 y$$

$$v = a_5 + a_6 x + a_7 y$$

in a region of pixels (x, y) near (x1, y1). For this region we use an 11×11 square window centered at (x1, y1). The parameters a_0 and a_1 model radiosity differences between f and g (in the vicinity of the match),

and the other six parameters define a 2d affine transformation between the pixel coordinates. In evaluating F we may restrict (x, y) to the integer pixel of grid of f, but we must compute g(u, v) at nongrid points (u, v). In our formulation we extend g by bilinear interpolation. Let u_0 = the greatest integer $\leq u$, v_0 = the greatest integer $\leq v$, $u_1 = u_0 + 1$, and $v_1 = v_0 + 1$, so that (u_0, v_0), (u_1, v_0), (u_0, v_1), and (u_1, v_1) are the grid points about (u, v). Then the interpolated value of g is given by

$$g(u, v) = (u - u_0)(v - v_0)\, g(u_1, v_1) + (u_1 - u)(v - v_0)\, g(u_0, v_1)$$
$$+ (u - u_0)(v_1 - v)\, g(u_1, v_0) + (u_1 - u)(v_1 - v)\, g(u_0, v_0)$$

Inserting this equation into that for F, the partial derivatives of F with respect to the parameters a_j may then easily be derived, and inserted into an iterative least squares procedure. This adjustment treats each pixel in the 11×11 window as an observation, and determines optimal values of the parameters $a_0, a_1, \ldots, a_7$. The values of $a_2, a_3, \ldots, a_7$ together with (x1, y1) are then substituted in the equations for u and v above to compute the refined location of (x2, y2).

References

[1] W. Forstner, A Feature Based Correspondence Algorithm for Image Matching, Int. Arch. for Photogrammetry and Remote Sensing, 26-3/3 (1986), 150-166.

[2] A. Grün, Adaptive Least Squares Correlation - A Powerful Image Matching Technique, South African Journal of Photogrammetry, Remote Sensing and Cartography 14 (1985), No. 3.

[3] M. Trajkovic and M. Hedley, Fast Corner Detection, Image and Vision Computing 16 (1988), 75-87.

[4] Z. Zhang, R. Deriche, O. Faugeras, and Q. Luong, A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry, Rapport de recherche 2273 (1994).